7. Computing Resources

This chapter includes computers and associated operating systems, peripherals and communication devices. The relationship of this chapter with the ITSG is shown in Figure 7-1.

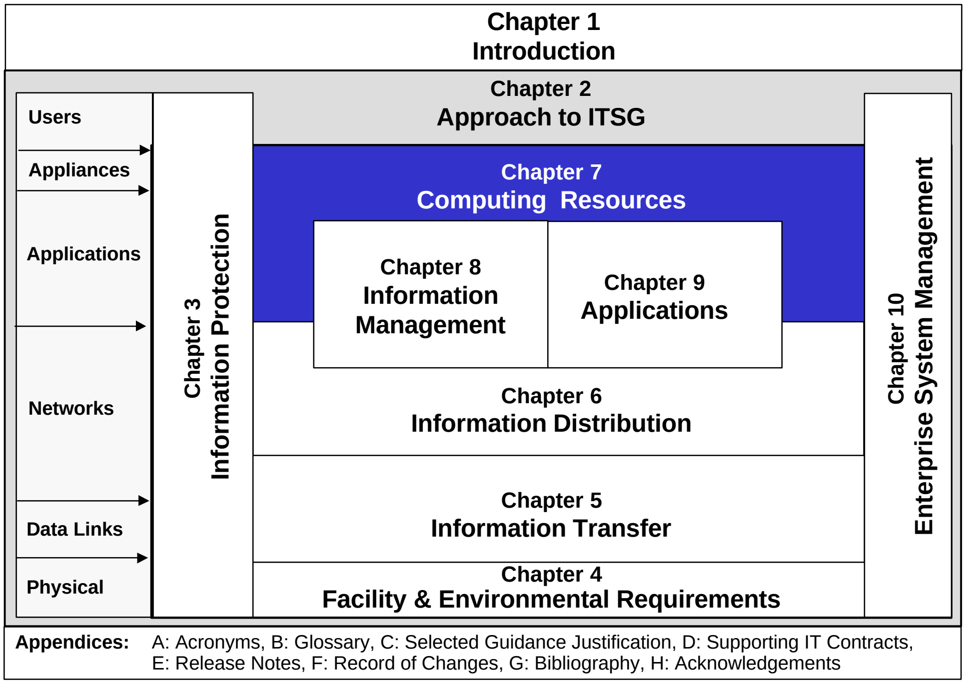


Figure 7-1. ITSG Document Map Highlighting Chapter 7, Computing Resources

7.1 Overview

The purpose of this section is to specify the computing resource standards that the Navy and Marine Corps should use for integrated, interoperable systems that meet the warfighter's minimum information processing requirements. Computing resource standards support the objectives of reducing cost and time of development, easing software integration and maintenance, and improving interoperability. As shown in Figure 7-2, computing resource technology is rooted in, but moves beyond the network technologies represented by the seven layer ISO/OSI model and the previous chapters.

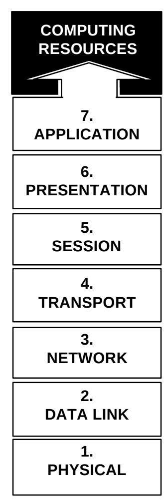


Figure 7-2. ISO/OSI Model and the Computing Environment

7.1.1 Background

The computing resource represents the user's window to the IT world (Figure 7-3). It encompasses the application servers, data servers and presentation clients that make up the three-tiered application architecture discussed in Chapter 9. Figure 7-3 depicts the network-centric model used in the ITSG in the context of the computer-centric DOD Technical Reference Model (TRM) (Figure 2-17). As shown, computing resources provide the connection between the operator, the network, applications and information.

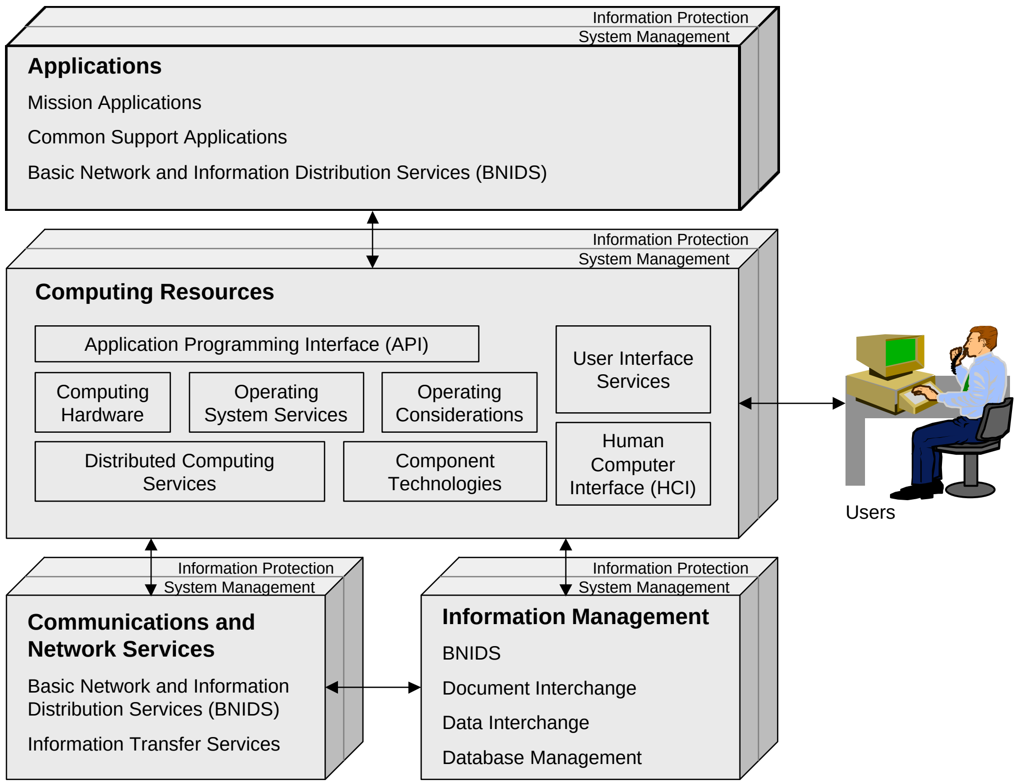


Figure 7-3. Computing Resources in the Context of the DOD Technical Reference Model

7.1.2 Scope of Computing Resources

Computing resources consist of operating system services and computing hardware, and may be based on the considerations of the external environment in which the computing resource is placed. The operating system services consist of an Operating System (OS), Application Programming Interfaces (API) and Human Computer Interface (HCI); they control the system resources in a single machine or distributed environment. The computing hardware includes a Central Processing Unit(s) (CPU), Input and Output (I/O) interfaces, main memory, buses and peripherals. The external environment considerations that affect computing resource selection are security, communications, real-time, and high availability. The computing resource provides the environment necessary to support application software. From the perspective of the application software, services are provided by the computing resource, whether the particular services are provided locally or remotely as part of a distributed system.

7.1.3 General Philosophy

The architecture needed to support a typical application consists of computers that perform as clients and servers. The servers host the primary application software and contain the processing power necessary to support multiple users. Servers also host the data needed to support the application. Figure 7-4 depicts the standard 3-tiered application architecture consisting of (1) an application server, (2) a data server, and (3) presentation clients. Configuration of these elements is more fully discussed in Chapter 9, “Applications.”

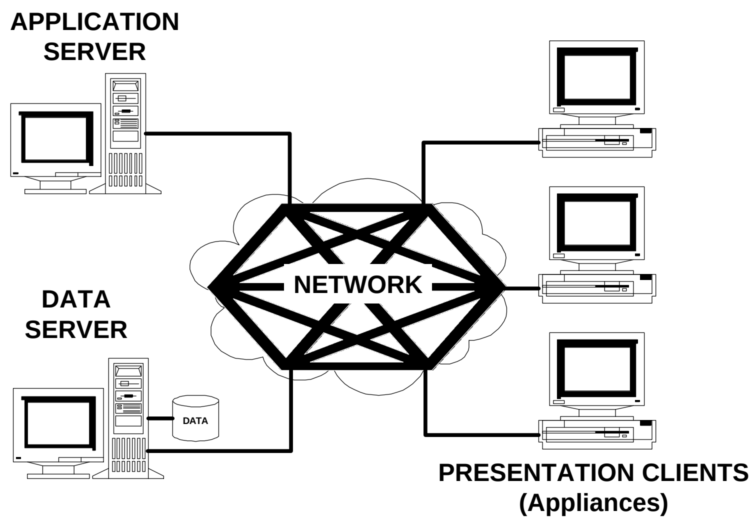


Figure 7-4. Three-Tiered Application Architecture

In the future, almost all application processing software will be hosted on the server computers. The clients will use presentation software that connects to the servers using a common interface[11]. At that time, client computers will likely be less expensive and tailored to the user's individual preference because application interoperability will not be a significant factor. Network computers are an example of personal workstations postured to support this application architecture at lower cost than today's personal computers.

Today, however, most application software is hosted on the client and interoperability among clients is a critical factor. Even within the client-server application architecture, application specific software resident on the client is still prevalent. This demands consistency of client workstations across the entire enterprise to achieve seamless trans-enterprise interoperability, which allows DON business process reengineering. Figure 7-5 outlines the rationale for the DON's client-server enterprise strategy.

Rationale for Heterogeneous Servers

- The server must be tailored to the specific application that may not be supportable by computers most prevalent in the marketplace.
- Many applications work well in their current computing environment and it is not cost effective to change.
- It is not practical to have all applications on a common server for multiple reasons including the need to maintain competition between computer developers and vendors.
- Encourages innovation by not restricting the type of computer used for the development of applications.

[11] Today, the presentation software is a network browser such as Netscape, Internet Explorer, Spry Mosaic, and HotJava.

Rationale for Homogeneous Clients

- Allows for a common, consistent user interface
- Maximizes interoperability
- Minimizes re-training required as users transfer to different organizations within the enterprise.
- Maximizes the ability to use common support and maintenance skills, parts and labor; thereby minimizing cost
- Maximizes portability of support for applications across the enterprise as well as portability of user skills
- Allows for economies of scale in both procurement (volume discounts) and support (more focused skill set for help desk personnel).

State of Application Technology:

- 1999/2000– Client-Server with client-hosted, application-specific software required: Highly dependent on the client's operating system.
- Future – Client-Server without the requirement for application-specific software on the client: dependent on the client's presentation software or network browser. Low dependence on the client's operating system.

Case for Windows NT as Common Client Operating System for 1999/2000

- Member of the Defense Information Infrastructure Common Operating Environment (DII COE)
- Has achieved and is in process of maintaining C2 evaluation criteria for trusted computer systems (Orange Book criteria).
- Supported by a large set of Government Off-The-Shelf (GOTS) software applications such as Defense Message System (DMS), Global Command and Control System (GCCS), Standard Personnel Support System (SPSS), and the Naval Tactical Command Support System (NTCSS)
- Supported by the largest selection of Commercial Off-The-Shelf (GOTS) software applications

Figure 7-5. Rationale for the DON enterprise client-server strategy

Driven by the current state of client-server technology, the general philosophy for implementing computing resources in the DON is the concept of homogeneous clients and heterogeneous servers. Homogeneous clients facilitate providing a consistent interface between the user and the system and make system support and maintenance less complex. Heterogeneous servers support the various computing requirements of applications needed to support DON enterprise missions. The same advantages that homogeneous clients enjoy can be achieved if servers are homogeneous as well. Independent of whether or not the server suite employed is heterogeneous or homogeneous, it is important that they perform their function transparently to the user (i.e., the user neither knows nor cares about the location, number, or vendor of the server being used.) Requiring servers to be homogeneous would restrict the introduction of new server technology which could choke innovation and prevent the enterprise from taking advantage of advances in computing such as massively parallel processors.

Current DON guidance is to develop all new applications such that the client can operate these applications at full capability using a Personal Computer (PC) running Windows NT. An important part of any migration to Windows NT is to identify and assess legacy and high performance computing environments. Many organizations cannot in the near term transfer their applications to Windows NT because of economic or performance issues. The DON CIO will support the connectivity of these legacy and high performance systems to the rest of the Navy and Marine Corps — concurrent with the move to Windows NT. Moreover, the ITSG will support all computing environments needed by Navy and Marine Corps organizations to perform their mission. For example, research and development organizations have a need to continue to expand the frontiers of computer science, and consequently should explore computer technology that offers potential for improving mission capability.

7.2 Computing Hardware

Computing Resources consist of many computing hardware components and configurations of these components. This section covers the various hardware components that make up a computing resource system and examines how these components are commonly configured.

7.2.1 Component Technologies

The major hardware components of Computing Resources are the Central Processing Unit (CPU), one or more backplane buses, main memory (both RAM and cache), Input/Output (I/O) interfaces, and peripherals. This section will examine each of these areas and provide guidance on the selection of these component technologies as part a computing resource system.

7.2.1.1 CPU

The CPU is the “engine” of the computer system and combined with the OS forms the core of the computing resource. Since the OS drives many decisions concerning the computer resource, a CPU that is compatible with the OS becomes an overriding factor in determining the type of CPU. Other than the OS, the main factors to consider in determining the type of CPU for the computer are processing speed (performance) and cost. For computing resources, such as servers and multiprocessors, scalability of the number of processors can be a significant factor in determining CPU.

Best Practices

Choose a CPU that is compatible with the OS of choice, has the performance to meet requirements, and is priced to fit the budget.

Recommended Implementations

	Current ITSG	Projected ITSG			
Not Recommended	**1999**	**2000**	**2001/2002**	**2003/2004**	**Emerging**
Clients & Servers & Special Purpose					
	Pentium II Pentium Pro Pentium AMD K6 Cyrix M2	Pentium II Pentium Pro AMD K6 Cyrix M2	Pentium ? AMD ? Cyrix ?	Pentium ? AMD ? Cyrix ?	Merced
Servers & Special Purpose					
	MIPS Alpha PA-RISC Ultra SPARC Power PC	MIPS Alpha PA-RISC Ultra SPARC Power PC	MIPS Alpha PA-RISC Ultra SPARC Power PC	MIPS Alpha PA-RISC Ultra SPARC Power PC	
Activities, Platforms, Operational Environments		All			

Table 7-1. CPU Recommended Implementations

- The Pentium Pro does not support Multi-Media Extensions (MMX) technology and therefore suffers a performance handicap when compared to the Pentium II, AMD K6 or the Cyrix M2. The Pentium II processor combines the features of the Pentium Pro processor with Intel's MMX technology resulting in the power and speed needed for intensive processing tasks.
- It is recommended that any new Pentium processor support MMX technology. Even on applications that are not graphics/multimedia intensive, there is a 8 – 15 % increase in performance over non-MMX capable PCs.
- Pentium substitutes like AMD K6 and CYRIX M2 are viable and may be considered. The Pentium II can be purchased with SD RAM which speeds up the processor by reducing the wait state from about 60 to 10 nanoseconds.
- The MIPS RISC processor is from the Silicon Graphics, MIPS Group. Alpha is produced by DEC. PA-RISC is from Hewlett-Packard. Ultra Sparc is a Sun Microsystems processor. The PowerPC is from IBM/Motorola. These systems are suitable as servers or special purpose workstations where PCs are not able to perform the required function.

7.2.1.2 Bus

The computer bus connects the different components of the computer resource together and allows them to pass data between them at high speeds. Computer resource configurations, such as COTS personal workstations, often limit or determine the type of bus that will be used. Often there are multiple buses connected together to allow for multiple types of component cards or to extend a non-expandable system bus. Considerations in determining the type of bus to use are: number and type of COTS products compatible with the bus architecture, number of parallel data bit lines, clock speed, and cost. Once the appropriate bus architecture is determined, an important computer resource factor becomes how many interface slots are available on the bus for component cards.

Best Practices

Use buses that provide the necessary performance economically and are compatible with the board level components that are needed to meet requirements. For buses that provide slots for component cards, use standard buses that are supported by multiple vendors providing compatible component cards.

Recommended Implementation

	Current ITSG	Projected ITSG			
Not Recommended	**1999**	**2000**	**2001/2002**	**2003/2004**	**Emerging**
ISA	PCI	PCI	PCI	PCI	
Microchannel	EISA				
	VME	VME	VME	VME	
	VXI	VXI	VXI	VXI	
	CardBus	CardBus	CardBus	CardBus	
Activities, Platforms, Operational Environments		All			

Table 7-2. Bus Standards

Notes

- Peripheral Connect Interface (PCI) bus is quickly gaining favor as the preferred system bus architecture. PCI provides the necessary throughput to support the high-end data rates required by many of today’s applications. Most COTS computers come with a PCI bus.
- EISA bus should be used only to accommodate legacy systems.
- Although VME bus is more popular, VXI bus offers a greater degree of standardization and therefore a greater degree of interoperability between vendor’s products. These are the buses of choice for embedded systems.
- CardBus is the new 32-bit high performance bus defined by the new PC Card Standard released by the PCMCIA standards body and trade association. The PC Card Standard replaces the outdated PCMCIA version 2.0 and version 2.1 standards.

7.2.1.3 Main Memory

Main memory is the storage warehouse of the computer where data and programs are stored for efficient processing. In the context of this section, main memory refers to cache and RAM. The main factor to consider in acquisition of a computer system is the quantity (in megabytes) of RAM. Other considerations are access speed, mounting design, and parity. Computer systems with too little memory run slowly, won't load, and crash often. Mounting designs today generally provide for Single In-line Memory Modules (SIMMS) with 72 or 168 pins. The older architecture SIMMS with 30 pins are generally slower and less efficient, and should be avoided to the extent possible.

Cache is usually hard wired to the motherboard and has a faster access time than RAM. Computer system caches of 512 KB or larger are generally satisfactory.

RAM can often be on a separate memory board and is used to store the OS, applications that are running, and data files. The amount of RAM needed for a computer system can vary with the environment and the OS. Servers generally need about an order of magnitude more RAM than personal workstations. For personal workstations with Windows 95, 32MB or more of RAM is recommended; for Windows NT, 64MB or more is recommended. For servers, use the Network Operating System (NOS) guidelines based on the environment. The three major factors used to determine the amount of RAM for a server are number of user connections, number of processes running, and amount of hard drive space.

Best Practices

Acquire enough memory, both cache and RAM, to allow computer performance to meet requirements.

Recommended Implementation

	Current ITSG	Projected ITSG			
Not Recommended	**1999**	**2000**	**2001/2002**	**2003/2004**	**Emerging**
32 pin RAM FPM	EDO SDRAM DRAM	EDO SDRAM DRAM	EDO SDRAM DRAM	EDO SDRAM DRAM	
Activities, Platforms, Operational Environments		All			

Table 7-3. Memory (RAM) Standards

Notes

While Extended Data Out (EDO) RAM is the current standard for mass market memory, a faster standard emerging is Synchronous DRAM (SDRAM). SDRAM is a memory architecture that incorporates many improvements over traditional DRAM technologies. SDRAM is a new technology that runs at the same clock speed as the microprocessor. The clock on the memory chip is coordinated with the clock of the CPU, so the timing of both the memory and the CPU are "in synch". This reduces or eliminates wait states, and makes SDRAM significantly more efficient than Fast Page Mode (FPM) or even Extended Data Out (EDO) memory.

7.2.1.4 Input/Output (I/O) Interfaces

I/O interfaces allow the computing resource to move data between the "outside world" and the CPU and main memory. Operations like loading a program or file from a floppy or CD, sending and receiving information over the LAN or WAN, or sending a document to the printer to get a hardcopy use I/O interfaces. Quite often the information is sent to or received from a peripheral, which is discussed in the next section.

Best Practices

Use I/O interfaces that use open access standards, support open device connections and are platform independent.

Recommended Implementation

	Current ITSG	Projected ITSG			
Not Recommended	**1999**	**2000**	**2001/2002**	**2003/2004**	**Emerging**
SCSI-1	SCSI-2	SCSI-2	SCSI-2	SCSI-2	XIO
SSA	SCSI-3	SCSI-3	SCSI-3	SCSI-3	IEEE 1394
Bus & Tag	FC-PH	FC-PH	FC-PH	FC-PH	
IDE	ESCON	ESCON			
	E/IDE				
	IPI	IPI	IPI	IPI	
	PC Card	PC Card	PC Card	PC Card	
	CardBus	CardBus	CardBus	CardBus	
	Serial	Serial	Serial	Serial	
	Parallel	Parallel	Parallel	Parallel	
Activities, Platforms, Operational Environments		All			

Table 7-4. I/O Interface Standards

Notes

- Fibre Channel (FC-PH) is emerging as a host-level interface standard for delivery of high I/O data transfers. FC-PH provides connections for workstation clustering, storage clustering and network-based storage concepts, parallel processing, load leveling, host-to-host or server-to-server communications, host- or server-to-mass storage communications, bulk data transfer, and multimedia. FC-PH is also being used as a system bus at the CPU and memory level as well as a way to cluster multiple systems similar to Non-Uniform Memory Access (NUMA). NUMA describes an architectural approach to clustering multiple systems such that distributed memory appears to the operating system as shared memory. This architectural approach allows a benefit to the user of a shared memory programming model with the scalability of a massively parallel processor's (MPPs) distributed memory model. FC-PH currently provides up to 1.062 GigaBits Per Second (Gbps) bandwidth using optical fiber.

- FC-PH-based storage subsystems will be supported on mainframes; therefore, the Enterprise Systems Connection or ESCON-based systems will be phased out.
- The Upper Layer Protocol (ULP) over FC-PH from the mainframe will initially deploy as SCSI-2 or SCSI-3 and migrate quickly to Intelligent Peripheral Interface (IPI). IPI will be deployed on large servers before it will be deployed on mainframes. Note that IPI is a protocol only; SCSI is both a protocol and an interface. FC-PH is an interface that can support various ULPs.
- For servers and higher end personal workstations SCSI is the predominant host-level interface for current disk and peripheral devices. As the demand for higher data transfer rates and expanded connectivity requirements increase, SCSI-2 and SCSI-3 over FC-PH in a PCI form factor will be the appropriate long-term direction.
- Enhanced IDE is the predominant peripheral interface for lower end personal workstations. SCSI is the current interface of choice for high-speed devices, such as high-capacity disk and tape drives and is the appropriate long-term direction for both the commodity level and high capacity devices.
- Personal Computer Memory Card International Association (PCMCIA) announced that PCMCIA cards are now referred to as PC Cards. The PC Card Standard defines a 68-pin interface between the peripheral card and the PC Card 'socket' into which it gets inserted. It also defines three standard PC Card sizes, Type I, Type II and Type III. All PC Cards measure the same length and width, roughly the size of a credit card. Where they differ is in their thickness. Type I, the smallest form factor, often used for memory cards, measures 3.3 mm in thickness. Type II, available for those peripherals requiring taller components such as LAN cards and modems, measures 5 mm thick. Type III is the tallest form factor and measures 10.5 mm thick. Type III PC cards can support small rotating disks and other tall components.
- As mentioned under "Bus" (Section 7.2.1.2), CardBus is a new standard introduced as an addendum to the PCMCIA PC Card standard. CardBus is a 32-bit bus mastering card operating at 33 Mhz transferring data at up to 132 MBytes per second. (The 16 bit PC Card bus data rate is 20 MBytes per second.) Like PC Cards, CardBus uses the 68-pin interface but operates at 3.3 volts versus the 5 volts used by PC Cards. CardBus slots are backward compatible with PC Cards. Card sockets can support PC Cards only or CardBus cards only, but not a mixture of the two.
- Serial I/O refers to standard serial interfaces such as RS 232, 422, 423, and 449.
- Parallel I/O refers to standard parallel interfaces such as micro-Centronix
- XIO is an emerging I/O solution based on Specialix' SI controller, and incorporating the high performance I/O processor and communication technology introduced in the RIO range. XIO is ideal for the 16-32 user system where sustaining high performance is essential.
- IEEE 1394, also known as "firewire", is a high speed serial I/O that supports data rates up to 400Mbps

7.2.1.5 Peripherals

Peripherals provide data access, input, storage, and connectivity for a computing resource. The number of peripherals available on the commercial market continues to explode, generally driven by processor speeds, memory/storage capacities and I/O speeds. Although there are many different types of peripherals, such as printers, facsimiles, modems, scanners, video cameras, microphones, speakers, etc., the main issue in specifying/procuring these items is the compatibility of their I/O interfaces with the computer (see Section 7.2.1.4) and application software. Apart from these compatibility issues, the major considerations for acquiring peripherals are cost and performance (which can include both speed and quality).

A major category of peripherals is static storage devices. As distinguished from main memory (covered in Section 7.2.1.3), static storage devices retain data when the power is off. The remainder of this section will discuss storage devices.

Best Practices

Use peripherals that support standard I/O interfaces and are platform independent.

7.2.1.5.1 Storage Device Standards

Storage refers to the capability to store information outside the central processor. For most computers, the predominant technology for storage has been magnetic disk and will remain so for the next few years.

Storage media is the physical material on which data is stored. The choice of media is usually determined by the application needs in terms of data accessibility, storage density, transfer rates, and reliability. Broad industry standards exist for 3.5″ magnetic disk, 4 mm digital audio tape (DAT), 8 mm helical tape, digital linear tape (DLT), ½″ tape, and compact disk-read only memory (CD-ROM). However, industry is in the process of agreeing on a standard for Digital Versatile Disk (DVD) which will able to read CD-ROMs as well as the new DVDs. Standards do exist for write once read many (WORM) optical and magneto-optical (MO) disks. Although the physical medium for optical technologies conforms to an open standard, the device’s recording format may not.

Archived data on outdated storage media (e.g. 5.25” floppy disks) should be transferred to media that is more current to avoid data being “trapped” on obsolete media that cannot be read by devices currently on the market.

As networks proliferate and storage requirements expand, storage technology that uses standard interfaces and promotes hardware and software supplier independence is necessary. This technology will enable us to take advantage of the open systems environment.

Best Practices

Implement storage technology and storage device media that use open access standards, support open device interface standards, and are platform independent.

Recommended Implementations

	Current ITSG	Projected ITSG			
Not Recommended	**1999**	**2000**	**2001/2002**	**2003/2004**	**Emerging**
5.25″ magnetic disk 5.25″ floppy disk 14″ WORM QIC (quarter inch cartridge) 3420 and 3480 Tape	3.5″ magnetic disk 3.5″ floppy disk 5.25″ and 12″ WORM 5.25″ MO ISO 13346 CD-ROM DVD 8 mm and ½″ helical tape 4 mm DAT DLT Tape 3490E cartridge tape	3.5″ magnetic disk 3.5″ floppy disk 5.25″WORM 5.25″ MO ISO 13346 CD-ROM DVD ½″ helical tape 4 mm DAT DLT Tape 3490E cartridge tape	3.5″ magnetic disk 3.5″ floppy disk 5.25″WORM 5.25″ MO ISO 13346 CD-ROM DVD ½″ helical tape 4 mm DAT DLT Tape 3490E cartridge tape	3.5″ magnetic disk 3.5″ floppy disk 5.25″WORM 5.25″ MO ISO 13346 CD-ROM DVD ½″ helical tape 4 mm DAT DLT Tape 3490E cartridge tape	2.5″ magnetic disk 3.5″ floptical Smart cards High capacity tape Virtual volume tape
Activities, Platforms, Operational Environments		All			

Table 7-5. Storage Device Media

Notes

- 8 mm helical tape is appropriate for backup and archive use; however, it is slow and unreliable for near-line storage solutions where frequent access is required. 4mm DAT offers a faster, more reliable solution to high-capacity and high-access applications.
- ½″ helical tape offers higher storage capacity than 4 mm or 8 mm; however, it is expensive, proprietary, and only Storage Technology supports it.
- 3490E square cartridge ½″ tape is predominantly a mainframe technology for off-line data storage.
- High capacity tape applies capacity multipliers of 2x-4x to current technology. Gains are achieved through increased density factors as well as media length (number of media feet per unit).
- Virtual Volume Tape will exploit tape capacity utilization by building multiple “virtual” tape volumes on a single physical tape media unit; as such, many small and otherwise inefficient tape consumers may be consolidated to achieve nearly 100% utilization of a tape resource.
- Smart cards are being used in Europe and Asia. Smart cards will be used predominantly as an intelligent storage media for providing services to individuals.

- Sony, Phillips, and Toshiba have jointly developed the Digital Video Disk (DVD, also known as Digital Versatile Disk) proposed standard for the CD-ROM industry. The standard allows for dual-sided as well as dual-layered implementations that will increase CD-ROM capacities significantly. In addition, transfer rates may improve to as high as 16 Mega Bits Per Second (Mbps).

7.2.1.5.2 RAID Technology

Redundant Array of Independent Disks (RAID) technology protects from data loss by providing a level of redundancy immediately within an array. The array contains removable disk drive modules that are automatically rebuilt in the event of a device failure without causing the system to shut down. When RAID levels other than 0 are used, no downtime is required to replace a failed disk drive. Data is continuously available while reconstruction of the failed disk occurs in the background.

Much of the benefit of RAID technology lies in its capability to off-load storage management overhead from the host system. To realize this benefit, RAID developers endow their array controllers with significant levels of intelligence. For instance, Adaptive RAID supports multiple RAID levels based on workload characteristics.

Best Practices

Choose the RAID level based on your specific need.

Recommended Implementation

Expiring ITSG	Current ITSG	Projected ITSG			Potential ITSG
Not Recommended	**1999**	**2000**	**2001/2002**	**2003/2004**	**Emerging**
	RAID 1,3,4,5,6 Just a bunch of disks	RAID 1,3,4,5,6	RAID 1,3,4,5,6	RAID 1,3,4,5,6	Adaptive RAID
Activities, Platforms, Operational Environments		All			

Table 7-6. Disk RAID Levels

Notes

For information and guidance for choosing specific RAID levels, see the web page at http://www/cis/ohio-state.edu/hypertext/faq/arch-storage/part1/faq-doc-84.html.

7.2.2 System Configurations

The hardware component technologies mentioned in Section 7.2.1 can be configured in many different ways to accomplish different tasks and meet different requirements. This section examines some of the common configurations (Personal Workstations, Servers, Enterprise Computers and Embedded Computers) with guidance on what component technologies to use for each configuration.

7.2.2.1 Personal Workstation

Personal Workstations (PW) are devices that contain at least one CPU (sometimes several) and provide a user interface, typically a GUI, as well as personal productivity tools, local data storage, and a flexible method for accessing and manipulating data. These PWs are commonly known as Personal Computers or PC's, desktop computers, portables (laptops or notebooks), or workstations and use one of several bus architectures. Low-end PWs are used primarily to support the general office work place. More powerful PWs are predominantly used in high-end compute applications such as Computer-Aided Design/Computer-Aided Manufacturing/Computer-Aided Engineering (CAD/CAM/CAE), application development, multimedia, and decision support data analysis presentation.

Also included in PWs are handheld computers – Personal Digital Assistants (PDAs), also referred to as Personal Information Managers (PIMs). Handhelds are computer systems that fit in a person's hand, and are extremely portable. Handheld systems also tend to have two types of input methods: pen or keyboard. The pen input can be used as digital ink, a mouse, or for handwriting recognition.

Best Practices

Combine components that provide flexible, scaleable, and easy-to-use personal workstations that support the Client/Server model of computing, data access, and multimedia. Allow for an external communication device such as a modem or network interface card.

Recommended Implementation

CPU	Bus	Memory	I/O	Peripherals
Pentium II	PCI	RAM≥64MB	E/IDE	HD>2GB
Pentium Pro		Cache≥512KB		3.5" floppy
Pentium				CD-ROM
AMD K6			SCSI-2,3	
Cyrix M2			PC Card	

Table 7-7. Personal Workstation (Portable) Hardware Configuration Guidance

CPU	Bus	Memory	I/O	Peripherals
Pentium II	PCI	RAM≥64MB	E/IDE	HD>2GB
	EISA	Cache≥512KB		3.5" floppy
				CD-ROM
AMD K6			SCSI-2,3	DVD
Cyrix M2			PC Card or	
			CardBus	

Table 7-8. Personal Workstation (Desktop) Hardware Configuration Guidance

Notes

- EISA bus should be used only to accommodate legacy systems.

Figure 7-6 provides a quick summary of the nominal specifications for Personal Workstation implementation.

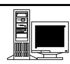
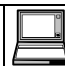
	 **Personal Workstation also used as a Server**	 **Desktop**	 **Laptop**
Processor	Pentium II Pentium Pro Pentium with MMX AMD K6 Cyrix M2 (200MHz Min.)	Pentium II Pentium Pro Pentium with MMX AMD K6 Cyrix M2	Pentium with MMX AMD K6 Cyrix M2
System Bus	PCI	PCI, EISA	PCI
Memory	256 MB SDRAM	64 MB SDRAM	64 MB SDRAM
Input/Output	E/IDE SCSI-3 PC Card X2 Serial Parallel Dual NIC	E/IDE PC Card X2 Serial Parallel Modem NIC (Modem and NIC can be on a PC Card)	E/IDE PC Card X2 Serial Parallel Modem NIC (Modem and NIC can be on a PC Card)
Storage	3.5” Diskette CD-ROM 4mm DAT 9 GB HDD	3.5” Diskette CD-ROM 2 GB HDD	3.5” Diskette CD-ROM 2 GB HDD
Operating System	Windows NT Server	Windows NT Client	Windows 95/98 or Windows NT Client

Figure 7-6. Personal Workstation Configuration Summary

7.2.2.1.1 Network Computers

Network Computers (NCs) are expected to be highly scaleable platforms, spanning a product range from the palmtop to the desktop. They must be attached to a TCP/IP-based network and interoperate with other nodes in the network. Though dependent on the network, NCs may offer some stand-alone functionality.

NCs are not intended to replace personal computers (PCs). Unlike PCs, however, NCs are designed for the network environment. Additionally, NCs are characterized by four specific attributes:

- Architectural neutrality

- Significantly lower total cost of ownership than PCs
- Lower entry price than the typical PC
- Significantly easier use and administration than PCs

In May 1996, Apple, IBM, Oracle Corp., Sun Microsystems, Inc., and Netscape Communications, Inc. announced a set of guidelines for developing low-cost, easy-to-use network computing devices. The NC Reference Profile 1 guidelines provide a common set of standard features and functions across a broad range of scaleable NCs. Architecturally neutral and intended to facilitate the growth of the network computing industry, NC Reference Profile 1 also protects investments made by customers, content providers, system providers, service providers, and application providers through industry-wide compatibility.

Microsoft and Intel have announced the specification for building a competing product to the NC, the NetPC, incorporating management capabilities and operating system features in a sealed PC case. The NetPC is a diskless or disked PC that has no data stored locally; its “identity’’ (user profile and configurations) is server-based.

Best Practices

Much has been said about the NC’s low cost and support benefits; however, these assertions have yet to be proven. Research and development organizations can purchase NCs to investigate potential utility.

7.2.2.2 Servers

Servers are computing resources that can be configured to support groups from small teams (work group servers) to entire sites (campus servers) to geographically dispersed organizations (enterprise servers). Work group servers provide support, such as directory, file, print, and authentication services, in a LAN environment. Campus servers may augment and, for many new applications, replace traditional mainframes. Selecting systems is important that address and support features and services of both systems management and reliability, availability, and serviceability. Campus servers often implement RAID storage technology to provide services with high reliability and availability (see Section 7.2.1.5.2).

There are different types of servers, performing more specific functions. Some of these server types include: database servers, multimedia servers, data push server, applications server, optical disk servers, and mail servers. A brief description of these servers follows:

Database Servers – Large databases make extensive use of disk space and processor power and typically require their own dedicated database server hardware. Increasingly these database servers have more than one CPU and more than one network interface channel to keep up with the demands placed upon them by large numbers of simultaneous users.

Multimedia Servers – If an organization has a large archive of audio or video data, or if it intends to distribute audio or video data in real-time, it will require a multimedia data server. Archives of audio and video data require huge amounts of disk space. Moving this data over the network (typically through streaming audio or video services and protocols) is also resource intensive. Both requirements typically require the dedicated use of a multimedia server.

Data Push Servers – If an organization implements a data push mechanism for distributing information to internal and/or external desktops, it will have to host that push function on a data push server. Depending on the number and volume of channels being pushed, the data push server software may reside on hardware that is also performing other server functionality (typically the web server).

Application Servers – One option for centralizing system management and reducing total cost of ownership in a network environment is to run the actual application programs on a few high-powered application servers rather than on the desktop clients. This is a common approach in UNIX-based computing environments through the use of the X Windows graphical interface to open display and control windows on one system for an application running on another system. A new variation of this approach brings current Microsoft Windows applications to systems that are under powered (old 286 or 386 PCs) or that are not running the Windows95/98 or WindowsNT operating system (UNIX workstations, Java Network Computers). This is made possible by running the applications on shared WindowsNT application servers and just sending the user interface over the network to the user's local system.

Optical Disk Servers – With the huge amounts of reference data available on CD ROM and the increasing availability of CD ROM writers for organizations to use to create their own archival data storage on CD ROM, there is a growing need to have multiple CD ROM disks online at once. A new class of hardware, the CD ROM Server, is a stand-alone network device that includes a large number of CD ROM drives which can be made available as shared disks to all users on a network.

Mail Servers – In an organization that has a number of networked PW clients, mail servers provide store and forward functions for electronic messages. These servers receive, store and distribute electronic messages and require a large amount of storage, proportional to the number of e-mail accounts they carry. They also require numerous network connections and modem connections to receive and distribute messages from/to users and other organizations.

Evaluate servers for their interoperability, portability, reliability, availability, serviceability, and scalability. The capability to easily upgrade processor performance or to add additional processors, disk storage, and communications support extends the life of the platform and enhances the return on investment.

Best Practices

Use hardware components that are standards based, primarily those that are compatible with interface bus standards. Select scaleable servers that can increase performance by adding components and by supporting standards-based network protocols. Through the use of a multiprocessing architecture, hardware should be scaleable and should enable parallel processing.

Recommended Implementation

CPU	Bus	Memory	I/O	Peripherals
Pentium II MIPS Alpha PowerPC PA-RISC Ultra SPARC	PCI VME VXI	RAM≥256MB* Cache≥2MB*	SCSI-2,3 FC-PH PC Card or CardBus IPI	HD>9GB* CD-ROM or DVD RAID 1,3,4,5,6 4mm DAT Tape

*Refer to server OS documentation to determine memory requirements

Table 7-9. Server Hardware Configuration Guidance

Notes:

PCI is a system bus architecture and interface standard that will be used to support I/O needs. Unlike SCSI, PCI is not an I/O interface. PCI will likely become the de facto open standard.

7.2.2.3 Enterprise Computers

Large-scale servers and enterprise computers represent the highest level of compute processing capabilities. This group of platforms consists of mainframes, symmetric multiprocessing (SMP) computers, and Massively Parallel Processor (MPP) computers.

Mainframe computers represent a greater share of the large-scale computing environment because they can support numerous terminals and large volumes of batch processing. Mainframes feature mature systems management tools, provide a high-availability operating environment, and support prevalent communications standards. Mainframes also represent an installed base that is entrenched within certain application environments. Within DoD, DISA is consolidating all mainframes into Defense Megacenters to use these expensive resources efficiently. The Standard Operating Environment (SOE) is the standard family of computer hardware and executive software comprising the mainframe operating environments within these Defense Megacenters. The focus of the SOE is on standardizing IBM/IBM compatible and UNISYS mainframe operating environments within the Megacenters.

Symmetric multi-processing (SMP) computers have multiple CPUs, are primarily standards based, and make frequent use of COTS components. SMP computers do not typically scale beyond 16 processors, although 64 processor servers are available. To increase scalability and availability, some SMP suppliers permit multiple computers (all from the same supplier) to be connected by a high-speed link to form what is commonly called a cluster. SMP computers support parallel processing; however, to use this feature, the operating system and the applications must also support parallel processing.

Massively Parallel Processors (MPPs) are generally associated with supercomputers. MPPs can potentially deliver computing power and I/O performance well beyond that of current mainframe and SMP architectures. MPPs promise to deliver high scalability, with some architectures containing thousands of processors. Typically, however, MPPs are more difficult to program and administer than SMPs.

A convergence is developing between traditional SMP and MPP architectures, resulting in hybrid architectures that borrow technologies from each other. These resulting hybrids show promise in becoming more viable for commercial applications than either of the traditional SMP or MPP architectures.

Best Practices

Avoid proliferation of costly, proprietary mainframe environments. Use open client-server architectures wherever possible. If mainframe environments are the only way to satisfy requirements, conform as much as possible to DISA's SOE.

When possible, avoid proprietary MPP architectures where the operating system and database are nonstandard and not open and where a shared memory programming model is not used.

Use enterprise computers with operating systems and hardware components that comply with IEEE's POSIX and TOG's XPG4 X/Open CAE specifications. Select scaleable servers that can increase performance by adding components and by supporting standards-based network protocols.

Recommended Implementation

CPU	Bus	Memory	I/O	Peripherals
TBD *	TBD *	TBD *	TBD *	TBD *

* To be determined by requirements

Table 7-10. Enterprise Computing Hardware Configuration Guidance

7.2.2.4 Embedded Computers

When COTS computers do not meet the requirements, components must be integrated to form a system. This often involves a design of board level components to perform the necessary functions. The major design consideration is the selection of a back plane bus that will allow all of the components to communicate with each other. This decision can impact dramatically the cost, both in the development and the logistics support of the system. Considerations in determining the bus include availability of Commercial Off-The-Shelf/Government Off-The-Shelf (COTS/GOTS) components and existing systems bus architectures (to reduce the logistics support costs). COTS/GOTS components should be considered first before custom designing and building the components.

Best Practices

When designing embedded computer systems, use industry standard buses and COTS/GOTS components to the maximum extent possible.

Recommended Implementation

CPU	Bus	Memory	I/O	Peripherals
Pentium II	VXI	TBD *	SCSI-2,3	Hard Drive
Power PC	VME		FC-PH	3.5” floppy
MIPS	PCI		IPI	CD-ROM
SPARC			ESCON	DVD
Alpha chip			E/IDE	
			PC Card or CardBus	
			Serial	
			Parallel	

* Varies widely based upon system requirements

Table 7-11. Embedded Computer Systems Hardware Configuration Guidance

Notes:

- Although VME bus is more popular and offers a wider variety of vendor products, VXI bus offers a greater degree of standardization and therefore a greater degree of interoperability among vendors’ products. These are the buses of choice for embedded systems.
- As the popularity of PCI bus rises and the number of COTS products increases, PCI becomes a more desirable architecture for embedded systems.

7.3 Operating System Services

Operating system services consist of the operating system (OS) and the interfaces the OS provides. These interfaces include a Human Computer Interface (HCI) which provides the “look and feel” of the computer system to the user and the Application Programming Interface (API) which provides the “look and feel” of the computer system to applications.

7.3.1 Operating Systems

The operating system (OS) schedules applications and other processes running on the central processing unit (CPU) and controls the exchange of data to and from the computing resource. The OS can reside on a single machine or can operate in a distributed environment. Although the user may not be aware of the single versus distributed environment, the factors to consider in procuring/designing the OS can vary significantly between the two environments. First, in “Local Computing Services” (Section 7.3.1.1), the single computing environment is considered. Next, in “Distributed Computing Services” (Section 7.3.1.2), the distributed computing environment is considered where the span of control of the OS extends beyond the local computing resource to multiple machines.

7.3.1.1 Local computing services

The local computing services are those OS services offered on a single machine. Most commercial OSs fit into this category. Considerations for determining the OS for a single computer environment depend mostly on the applications that will be running on the local machine and the connections needed to communicate with peripherals and the outside world.

7.3.1.1.1 Personal Workstation

As described in Section 7.2.2.2, Personal Workstations (PW) are devices that contain at least one CPU (sometimes several) and provide a user interface, typically a GUI, as well as personal productivity tools, local data storage, and a flexible method for accessing and manipulating data. When purchased off the shelf, PWs will come packaged with an OS. Another OS can be negotiated with the vendor; however, this may mean an added cost.

The major consideration for OS selection is the mix of applications that are intended to run on the PW. Interoperability at the applications level is very important and investigation into each application package is needed to determine the extent of interoperability with other application packages, both on the local PW and within the user's computing community. Another important consideration is communications with other computer resources, although most OSs allow for this. Currently, an important paradigm is the client/server model. Again most OSs allow for this; it is the application software that executes this model. Therefore, the important consideration here is to ensure that the OS is compatible with applications that support the client server model.

NCs support a common Java-based programming environment that enables network-resident applications, as well as stand-alone applications, to be executed on them. The NetPC will run the current Windows OSs (95/98 or NT).

Best Practices

Combine components that provide flexible, scaleable, and easy-to-use personal workstations that support the Client /Server model of computing, data access, and multimedia. Use operating systems that comply with DII COE for long-term usability.

Recommended Implementation

Expiring ITSG	Current ITSG	Projected ITSG			Potential ITSG
Not Recommended	**1999**	**2000**	**2001/2002**	**2003/2004**	**Emerging**
MS Windows 3.1 and older OS/2 MacOS	Windows NT Workstation Windows 95/98 (non-critical hosts and laptops)	Windows NT Workstation	Windows NT Workstation	Windows NT Workstation	Object-oriented OSs JavaOS Microkernel OSs
Activities, Platforms, Operational Environments		All			

Table 7-12. Personal Workstation Operating Systems

Notes:

- Migration to Windows NT is the appropriate direction in 1999. If business reasons justify a delay in migrating to Windows NT, Windows 95/98 is an acceptable operating system to use through 1999 if currently installed for non-critical systems. Avoid Windows 3.11 and older.
- Sun Microsystems is currently shipping JavaOS.

7.3.1.1.2 Servers

As described in Section 7.2.2.3, servers provide support, such as file, mail, and print services, in a LAN environment. A server's OS is often known as a Network Operating System (NOS).

Major considerations for server OSs are compatibility with client software, network support (at a minimum must support TCP/IP), and server type (applications server, database server, etc.).

World Wide Web (WWW) servers are often supplied as a bundle of hardware, operating systems, and Web-server software. The Web-server software is not an actual operating system; however, Web servers are being used as small to medium sized servers. WWW servers vary in size and complexity, according to the environment where they are deployed.

Best Practices

Use operating systems that comply with DII COE, TOG's X/Open CAE, or the Institute of Electrical and Electronics Engineer's (IEEE's) POSIX specifications. Operating systems should be scaleable, multitasking, multithreaded and enable parallel processing. It is also beneficial if servers are homogeneous with clients. For WWW servers, select products that support a secure version of the HTTP and provide the ability to scale up as volume increases.

Recommended Implementation

Expiring ITSG	Current ITSG	Projected ITSG			Potential ITSG
Not Recommended	**1999**	**2000**	**2001/2002**	**2003/2004**	**Emerging**
NetWare 3.x or less VINES	Windows NT Server UNIX 95 (X/Open CAE) POSIX NetWare 4.1 or later	Windows NT Server UNIX 95 (X/Open CAE) POSIX NetWare 4.1 or later	Windows NT Server UNIX 95 (X/Open CAE) POSIX NetWare 4.1 or later	Windows NT Server UNIX 95 (X/Open CAE) POSIX	Microkernel OSs
Activities, Platforms, Operational Environments		All			

Table 7-13. Server Operating Systems

Notes:

- SPEC 1170 and the XPG4 specifications are now included within the X/Open CAE specifications.
- Compliance with X/Open CAE and POSIX standards allows for increased interoperability of hardware components and facilitates application portability. Typically the brand, such as X/Open CAE, UNIX 93, and UNIX 95, applies to a combination of platform and operating systems. For example, some Xopen CAE-based, profile-branded platforms include HP-UX 9.X, running on HP 9000 series processors and UnixWare 2.0, the later running on Intel 486 and Pentium platforms.

7.3.1.1.3 Enterprise Computers

As mentioned in Section 7.2.2.4, enterprise computers represent the highest level of computer processing capabilities. Proprietary OSs and custom developed applications are common for these platforms. When determining the OS of enterprise computers, determine the extent to which the applications will need to communicate with other applications or whether the applications will need to run on other systems. If the applications/system will stand-alone and not interact with other systems, then OS should be chosen based on functionality (e.g., development environment, application programming interfaces, scheduling, etc.). If interactions with other systems will occur, ensure the OS is compatible with those systems (e.g., DISA mainframe systems will require compliance with their Standard Operating Environment.).

Best Practices

Use enterprise computers with operating systems that comply with X/Open CAE and IEEE's POSIX specifications. If mainframe environments are the only way to satisfy requirements, conform as much as possible to DISA's SOE.

Recommended Implementation

	Current ITSG	Projected ITSG			
Not Recommended	**1999**	**2000**	**2001/2002**	**2003/2004**	**Emerging**
	X/Open CAE POSIX OS/390 MVS, VM TPF	X/Open CAE POSIX OS/390 VM TPF	X/Open CAE POSIX OS/390 VM TPF	X/Open CAE POSIX OS/390 VM TPF	Microkernel OSs
Activities, Platforms, Operational Environments		Special Purpose Shore Information Producers			

Table 7-14. Enterprise Computer Operating Systems

Notes:

- SPEC 1170 and the XPG4 specifications are now included within the X/Open CAE specifications.
- Product branding for standards such as X/Open CAE, UNIX 93, and UNIX 95 normally applies to a combination of platform and operating systems. For example, some X/Open

CAE-based, profile-branded platforms include HP-UX 9.X, running on HP 9000 series processors.

- Most SMP computers and MPP computers use the UNIX operating system. Two of the major MPP suppliers have proprietary operating systems designed to interface with their own proprietary databases. These proprietary systems have been optimized to address performance issues in specific applications such as large complex queries of large databases.
- MVS and Virtual Machine (VM) are supported when migrating to an open, standards-based platform. Open Edition (Open MVS and Open VME) assists in this migration by providing standards-based tools and capabilities. MVS has been branded recently for base X/Open CAE.
- Transaction Processing Facility (TPF) is an IBM operating system designed to meet the needs of applications that support a large terminal base and require high transaction rates, quick response time, and high availability. TPF continues to improve usability by offering enhanced systems management, ISO-C, TCP/IP, and, eventually, DCE support and POSIX interfaces.
- When system requirements dictate a mainframe solution, use open system technologies whenever possible. Avoid proprietary application support technologies whenever feasible.

7.3.1.2 Distributed Computing Services

Distributed computing services are those application platform technologies that enable a Distributed System Architecture (DSA). A DSA is one in which multiple processing platforms participate cooperatively to accomplish an IM task. Conceptually, DSAs can range in size and scope from a set of dedicated microprocessor systems a few inches apart in an autonomous vehicle to an aggregate of supercomputers spread throughout the country working as a fluid dynamics analyzer. In every case, a protocol is required to enable the individual computers to cooperate. This section currently addresses DSAs connected through standard networks; bus-connected systems are deferred to a future update.

There is no single agreed-upon list of distributed computing services because differing system requirements dictate differing enabling technologies. For example, an application that refers its complex physics analyses to a vector processor may have no need for a time-synchronization service, while an accounting transaction processing system would. To provide guidance, an inclusive but not exhaustive list of services is provided. The services include:

- Remote Procedure Call (RPC)
- Security services
- Directory services
- Time services
- Threads service
- File service
- Object services

The first six services are included in the Distributed Computing Environment (DCE), as defined by TOG. Specifications for open object services have been developed by the Object Management Group (OMG) and are elements of the Common Object Resource Broker Architecture (CORBA).

These standards form the basis for creating heterogeneous distributed applications in both procedure-oriented and object-oriented development environments.

Finally, Inter-Process Communication (IPC) capabilities are required to implement DSAs. In designing a DSA application, candidate IPC technologies should be evaluated carefully for capability, performance, cost, and portability.

7.3.1.2.1 Remote Procedure Call

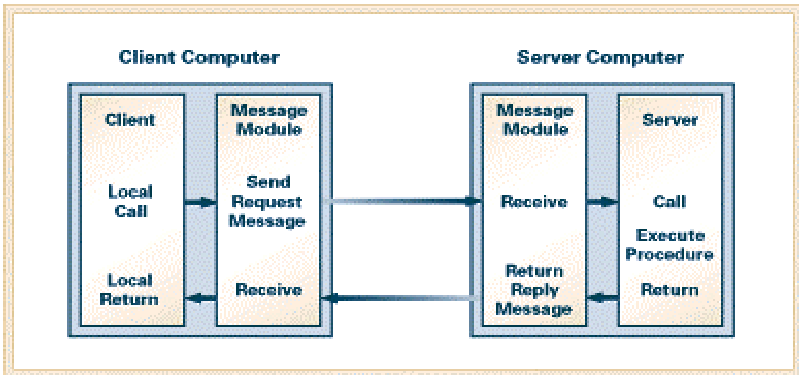


Figure 7-7. Remote Procedure Call

Remote procedure call (RPC) supports DSA designs by enabling a local program to execute procedures located on other computers in the network (Figure 7-7). The local program invokes remote procedures in much the same way as local procedures. Parameter passing between the local and remote procedures is supported. The local program does not need to know the location of the remote procedure; this information is acquired through directory services.

RPC technology can simplify the conversion of an existing monolithic application to a client/server design. To minimize the need for changes to source code, RPC tools can employ a precompiler utility to process Interface Definition Language (IDL) statements, which generate the source code that ensures compatibility between the client and the server.

Best Practices

Use DCE RPC and TxRPC as they become available. If DCE RPC is inappropriate for a specific implementation, use Open Network Computing (ONC) RPC or Netwise RPC.

Recommended Implementation

	Current ITSG	Projected ITSG			
Not Recommended	**1999**	**2000**	**2001/2002**	**2003/2004**	**Emerging**
	DCE RPC TxRPC ONC RPC TransAccess RPC	DCE RPC TxRPC ONC RPC TransAccess RPC	DCE RPC TxRPC ONC RPC TransAccess RPC	DCE RPC TxRPC ONC RPC TransAccess RPC	ISO RPC 10148
Activities, Platforms, Operational Environments		All			

Table 7-15. Remote Procedure Call

Notes:

- The DCE RPC, developed by TOG, has been adopted as an Open Group Common Application Environment standard and is the basis for the RPC standard being developed by ISO. The close integration the DCE RPC and the remaining DCE services makes DCE RPC more capable than other RPC technologies.
- TxRPC is a TOG standard that extends the DCE RPC to support transactional semantics for use in a distributed transaction processing (DTP) environment.
- ONC RPC, an older de facto RPC standard developed by Sun Microsystems, Inc., is UNIX-based and uses Transmission Control Protocol/Internet Protocol (TCP/IP). This RPC is simpler in design and more mature than DCE RPC. ONC RPC does not include the sophisticated distributed services that are integrated with DCE RPC.
- The TransAccess RPC, now owned by Proginet, is a mature but proprietary product based upon ONC RPC. Netwise extends the ONC RPC to support both Systems Network Architecture (SNA) logical unit (LU) 6.2 and NetWare Internetwork Packet Exchange/Sequenced Packet Exchange transport protocols. Netwise RPC enables Customer Information Control System (CICS), Information Management System (IMS), time share option, batch, and NetView programs on a Multiple Virtual Storage (MVS) platform to exchange data with RPC programs on other platforms.
- ISO RPC 10148 is an emerging standard.

7.3.1.2.2 Security Services

Security services authenticate the identities of users, authorize controlled access to distributed resources, and provide user and server account management. Security facilities are described fully in Chapter 3.

7.3.1.2.3 Directory Services

Directory services are used to locate distributed resources that reside on a network. Details regarding directory services can be found in Chapter 6.

7.3.1.2.4 Time Services

As stated in Section 6.6.1, time services are established to ensure consistency and accuracy of time and dates across distributed systems. Details regarding time services can be found in that section.

7.3.1.2.5 Threads Service

Threads are independent program elements that, taken together, accomplish the overall program objective. A threaded application can make effective use of a multi-processor computer because different threads can operate simultaneously on different processors. Assigning threads to processors is a function of the operating system. In a DSA, separate networked computers can work together as a larger multi-processor system. Threads service provides an operating-system-like capability to utilize this kind of concurrency in a DSA design.

Best Practices

Use DCE to acquire threads service. Each service implemented in DCE is internally integrated with threads service, giving the designer networked concurrency capability without including it as an application program element.

7.3.1.2.6 File Service

File service provides controlled access to files across a network. File service makes a variety of design options feasible, ranging from NCs and diskless personal workstations to geographically distributed databases.

Different file service products seldom interoperate. Since different file services have been available from different vendors for some time it is likely that incompatible legacy products are now in use. Further, file service products tend to be bundled with operating systems and network operating systems. Establishing a common file service in a heterogeneous DSA environment often requires adding new file service software to at least some systems that already incorporate another file service.

File services use network transport protocols to move data between local and remote computers. Network transport protocols have different characteristics that are inherited by services built upon them. For example, Network File Service (NFS) was originally designed to use User Datagram Protocol (UDP), a low-overhead but unreliable IP-based protocol that made NFS better suited to local-area rather than wide-area networks. NFS on TCP/IP provides greater data integrity in unreliable networking situations.

Access controls and data protection associated with different file service products vary widely. Some products may require add-on user identity services; some transfer clear data (text or binary) across networks. Other products integrate user authentication and encryption options with the basic file service capability.

Best Practices

Use DCE Distributed File Service (DFS).

Recommended Implementation

	Current ITSG	Projected ITSG			
Not Recommended	**1999**	**2000**	**2001/2002**	**2003/2004**	**Emerging**
	DCE DFS	DCE DFS	DCE DFS	DCE DFS	
Activities, Platforms, Operational Environments	All				

Table 7-16. File Service

Notes:

- Avoid file services that are specific to a particular operating system or network operating system.

7.3.1.2.7 Object Services

Object service provides capabilities similar to DCE using an object-oriented paradigm. The primary object service is called an Object Resource Broker (ORB). ORBs are middleware products used to implement object-oriented DSAs. In an ORB environment, a program will invoke a class method belonging to a particular object. The ORB receives the request, resolves it, and returns the results to the invoking program. To resolve the request, the ORB must locate an object supporting the method, invoke it, pass the necessary parameters to the method, and receive the results. The application program doesn't need to know the location of, nor accommodate communication with, any external object; objects can be locally implemented or reside on remote networked computers.

The predominant organization in developing ORB standards has been the Open Management Group (OMG). The Common Object Resource Broker Architecture (CORBA) is the underlying standard for several different commercially-available ORB implementations. OMG's Internet InterORB Protocol (IIOP) is used to assure interoperability among CORBA implementations.

Microsoft Distributed interNet Architecture (DNA) has been proposed as an alternative to the CORBA track. DNA includes the Distributed Component Object Model (DCOM), ActiveX, Active Directories, and several other elements. At present, DNA includes proprietary Microsoft products, limiting its portability.

Best Practices

Choose an appropriate CORBA implementation. Avoid proprietary ORBs that limit portability and opportunities for heterogeneity.

Recommended Implementation

	Current ITSG	Projected ITSG			
Not Recommended	**1999**	**2000**	**2001/2002**	**2003/2004**	**Emerging**
	CORBA 2.1	CORBA	CORBA	CORBA	OMG-submitted ISO
Activities, Platforms, Operational Environments		All			

Table 7-17. Object Service

Notes:

- Commercial CORBA implementations with demonstrated IIOP functionality are available from Digital Equipment Corporation, DNS Technologies, Expersoft, Fujitsu, Hewlett-Packard, IBM, ICL, IONA, Siemens Nixdorf, SunSoft, Tandem, and Visigenic.
- Other compliant products are likely to become available.

7.3.1.2.8 Inter-Process Communication

Inter-Process Communication (IPC) enables the exchange of data between application programs, procedures, or objects. Using IPC technology, programs can exchange data within a single computer or between computers over a network. Each IPC technology provides an API. The two types of IPC technologies are transport dependent and transport independent.

Transport-dependent IPC technologies are generally based on a single transport protocol or operating platform, providing a low-level API that has transport-dependent syntax. These characteristics inhibit application portability. Avoid transport-dependent IPC technologies when possible. Examples of transport-dependent IPCs are Microsoft’s Named Pipes, Novell’s transport layer interface, and IBM’s advanced program-to-program communication (APPC) and Common Programming Interface-Communication (CPI-C).

Transport-independent IPC technologies generally operate independent of transport protocols and operating platforms. These technologies provide a high-level API that has transport-independent syntax, is easy to use, and enables application portability. These technologies include RPC and ORB, above, and interprocess messaging.

Interprocess messaging is a middleware technology that uses message passing and message queuing to provide peer-to-peer asynchronous communication between programs. Messaging is a relatively mature technology that has been widely used for distributed applications involving high transaction rates in the banking, stock market, and airline industries. Few standards exist, however, for portable messaging APIs or interoperable messaging protocols.

7.3.1.2.8.1 Message Passing

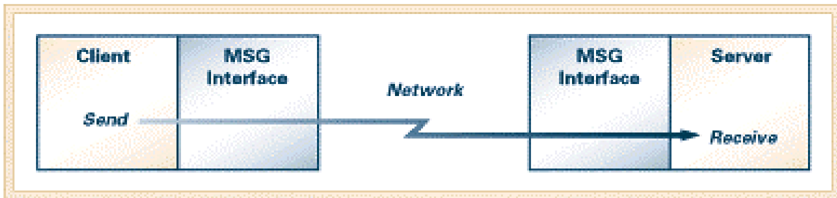


Figure 7-8. Message Passing

The foundation of messaging technology is a message-passing model in which local application programs call an API that can have as few as four verbs: open connection, send, receive, and close connection (Figure 7-8). In a client/server message-passing model, a client sends a request to the server in the form of a message. The server receives the message and processes the request. The server often creates a new message containing the reply and sends this reply message to the client.

Best Practices

Use standards-based products as they become available. Avoid transport-dependent products.

7.3.1.2.8.2 Message Queuing

In many implementations of messaging middleware, the basic message-passing model is enhanced with a message queue to provide a buffer for storing messages that have been sent and are waiting to be acted upon. In this enhanced model, the send verb of the API becomes a put-on-queue operation, and the receive verb becomes a get-from-queue operation (Figure 7-9).

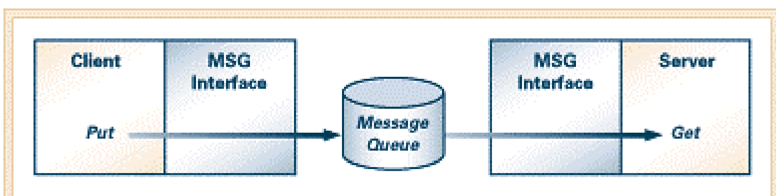


Figure 7-9. Message Queuing

In contrast to other IPC technologies, message queuing is inherently connectionless. In many message-queuing implementations, no direct connection is ever established between the application client and application server. With message queuing, the sender and receiver do not need to be simultaneously available to communicate, nor does the network need to be available directly between the sender and receiver. This capability to support discontinuous communication makes message queuing more tolerant of unreliable networks than other IPC technologies.

Message queuing has other characteristics that can be advantageous in specific application environments:

- Senders and receivers are not required to know each other.
- Messages can be processed in different sequences.
- Persistent queues can guarantee message delivery.

- Shared queues can support load balancing or parallel processing.

A proposed ISO draft standard defines a model, an application layer service definition, and a protocol specification for message queuing. If adopted, this ISO standard may promote interoperability between message-queuing products.

A proposed TOG draft standard defines a message-queuing API. If adopted, this TOG standard may promote source code portability between message-queuing products.

Best Practices

Use standards-based products as they become available. Avoid transport-dependent products.

7.3.2 Application Programming Interface

The OS or a shell on top of the OS will provide an interface by which application programs can exchange information with and access the computing resource. This interface, called the Application Programming Interface (API), defines the services that are available to applications. Having a common API across systems or organizations will promote the portability of application software written for that API. Commercial OSs (e.g. Windows NT) have an API that commercial application vendors build to. If a system is not used for application development, APIs are of little consideration in the procurement of a computing resource. The main consideration for those systems is the interoperability of the data among various applications (e.g., can a Word file be read and edited by a WordPerfect application?)

A key component of the DON application systems architecture will be a set of platform-independent APIs. Application platform differences will be “masked” by software for functional compensation and API translation. As application software is ported to new environments, high-level APIs and software to compensate for differences in the underlying products must be provided. To address these issues, the application systems architecture must supply design guidelines and APIs that will provide the required level of portability with minimum effort. An example of this is the migration of the Joint Maritime Command Information System (JMCIS) from UNIX to Windows NT.

Three dimensional (3D) graphics applications are becoming popular because they provide a more realistic representation of an information space and contribute to the development and maintenance of situation awareness by the warfighter. Because of 3D applications’ reliance on hardware and software architectures to support numerically intensive calculations and the rapid changes in these architectures, standards for 3D APIs are in a state of development by several organizations. The APIs for 3D computer graphics can be divided into those targeted toward the engineering and business requirements and those that support the video game environment. The 3D API for video games is a competitive environment with many players and few people willing to risk supporting a single API.

The APIs for the engineering and business environment are more mature. PHIGS, OpenGL, and HOOPS are currently competing in this market. Other 3D APIs (such as Open Inventor) exist in this market, but they tend to support slightly different goals.

Best Practices

The computing resource services should be accessed by applications through either the standard Win32 APIs, Portable Operating System Interface (POSIX), or TOG Single UNIX specification, version 2. Not all computing resource services are required to be implemented, but those that are used shall comply with the standards listed below.

The following standards are recommended:

- Win32 APIs, Window Management and Graphics Device Interface, Volume 1 Microsoft Win32 Programmers Reference Manual, 1993 or later, Microsoft Press.
- ISO/IEC 9945-1:1996, Information Technology - Portable Operating System Interface for Computer Environments (POSIX) - Part 1: System Application Program Interface (API)[C language] (The C Language is part of the formal title of this standard. It denotes the language used to define the standard.) (Mandated Services)
- ISO/IEC 9945-1: 1996:(Real-time Extensions)
- ISO/IEC 9945-1: 1996: (Threads Extensions)
- ISO 9945-2: 1993, Information Technology - Portable Operating System Interface for Computer Environments (POSIX) - Part 2: Shell and Utilities, (as profiled by FIPS PUB 189: 1994)
- IEEE 1003.2d: 1994, POSIX - Part 2: Shell and Utilities - Amendment: Batch Environment
- IEEE 1003.5 - 1992, IEEE Standard for Information Technology - POSIX Ada Language Interfaces - Part 1: Binding for System Application Program Interface (API)
- IEEE 1003.5b - 1996, IEEE Standard for Information Technology - POSIX Ada Language Interfaces - Part 1: Binding for System Application Programming Interface (API) - AMENDMENT 1: Real-time Extensions
- Single UNIX Specification, Version 2, API, The Open Group

7.3.3 Human Computer Interface

The Human Computer Interface (HCI) refers to the way a user interacts with a computer, including the display and manipulation of information as well as the user’s method of input and navigation within a system. HCIs are not limited to a visual display of information on a screen. Selection and use of I/O devices, such as the mouse, touch screen, and data glove, are considerations in the design of an effective HCI.

7.3.3.1 HCI Style Guides

An HCI style guide is a document that specifies design rules and guidelines for the look and behavior of the user interaction with a software application or a family of software applications. The style guide represents “what” user interfaces should do in terms of appearance and behavior, and can be used to derive HCI design specifications which define “how” the rules are implemented in the HCI application code.

Figure 7-10 illustrates the hierarchy of style guides to maintain consistency and good HCI design within the DON. This hierarchy, when applied according to the process in the DoD HCI Style Guide, provides a framework that supports iterative prototype-based HCI development. The process starts with top-level general guidance and uses prototyping activities to develop system-specific design rules.

The interface developer shall use the selected commercial GUI style guide, refinements provided in the DoD HCI Style Guide, and the appropriate domain-level style guide for specific style decisions along with input of human factors specialists to create the system-specific HCI.

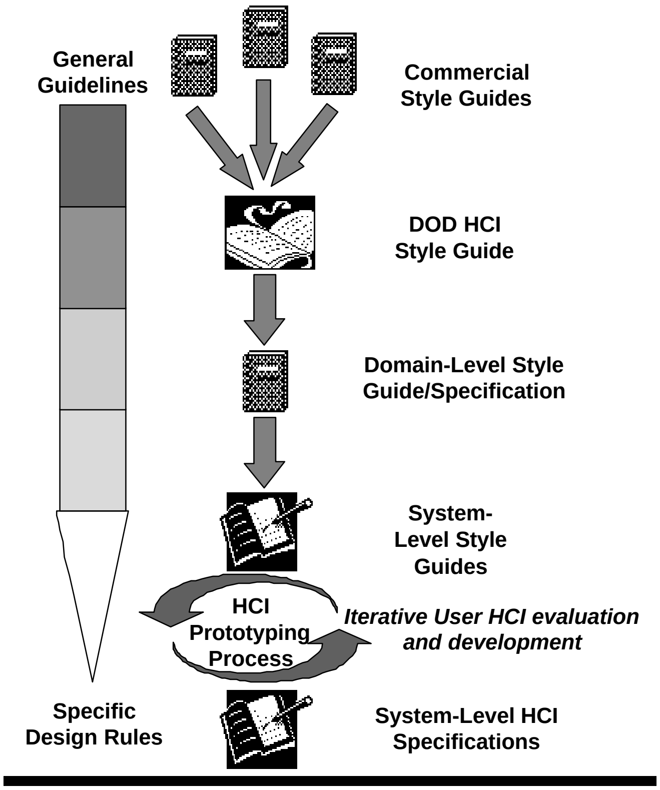


Figure 7-10. HCI Development Guidance

Most Web-based interfaces use Hypertext Markup Language (HTML) to describe the structure of the information they contain. The next version of the DoD HCI Style Guide and the User Interface Specifications for the Defense Information Infrastructure (DII) are expected to address HTML-based interfaces. The next version of the user interface specification for the DII addresses MS Windows based interfaces.

7.3.3.2 HCI Development

For many systems, COTS products will be used exclusively and there will be no development of an HCI. For some other systems, applications will be developed for a commercial OS, but there will also be no development of an HCI. Rather the HCI that is part of the OS will be used and will be quite satisfactory. Still other systems may have no HCI at all but rather get input from sensors and provide outputs to controller devices. For the remaining systems, special requirements will necessitate the development of an HCI.

This section provides guidance for HCI development and implementation in DON automated systems. The objective is to standardize user interface design and implementation options thus enabling DON applications within a given domain to appear and behave consistently. The standardization of HCI appearance and behavior within the DON will result in higher productivity, shorter training time, and reduced development, operation, and support costs. The following sections include specific guidance regarding the style guide hierarchy levels.

7.3.3.2.1 Character Based HCIs

For all DON automated systems, the near-term goal is to convert character-based interfaces to GUIs. Although GUIs are the preferred user interface, some specialized devices may require use of character-based interfaces due to operational, technical, or physical constraints. These specialized interfaces shall be defined by domain-level style guides and further detailed in system-level user interface specifications. In order to present a consistent interface to the user, application software shall not mix command line user interfaces and GUIs.

Best Practices

The following is mandated for systems with a requirement for a character-based interface:

- DoD HCI Style Guide, TAFIM Version 3.0, Volume 8, 30 April 1996.
- While not mandated, additional guidance for developing character-based interfaces can be found in ESD-TR-86-278, Guidelines for Designing User Interface Software (Smith and Mosier 1986).

7.3.3.2.2 Commercial Style Guides

When developing DON automated systems, the graphical user interface shall be based on one commercial user interface style guide. Hybrid GUIs that mix user interface styles (e.g., Motif with Windows) shall not be created. A hybrid GUI is a GUI that is composed of tool kit components from more than one user interface style.

Best Practices

A commercial GUI style shall be selected as the basis for user interface development.

The following is mandated for a Microsoft Windows-based environment:

- The Windows™ Interface Guidelines for Software Design, Microsoft Press, 1996 (Defines the Windows 95 and Windows NT 4.x HCI).

For UNIX workstations, the style guide corresponding to the selected version of Motif is mandated:

- Open Software Foundation (OSF)/Motif Style Guide, Revision 1.2 (OSF 1992)
- TriTeal Enterprise Desktop (TED) 4.0 Style Guide and Certification Checklist, Carlsbad, CA: TriTeal Corporation, 1995.

7.3.3.2.3 Enterprise-level Style Guides

The DoD HCI Style Guide focuses on elements of user interface design not normally addressed in a commercial style guide and concentrates on DoD-specific functional areas such as display of security classification markings, mapping display and manipulation, decision aids, and embedded training. The DoD Style Guide is intended to provide guidelines on good HCI design, it represents DoD policy and all DoD information systems are required to comply with its provisions.

Best Practices

The following guideline is mandated:

- DoD HCI Style Guide, TAFIM Version 3.0, Volume 8, 30 April 1996.

7.3.3.2.4 Domain-level Style Guides

Domain-level style guides provide detailed guidance that addresses the requirements of a particular domain (e.g., C4ISR) as defined by a DoD organization (e.g., joint, individual service, or agency). Domain-level style guides reflect the consensus of the organization on the look and feel they want to provide in their systems. Over time, it is expected that DoD organizations will develop and publish domain-level style guides for directing the HCI design efforts for their systems. The domain-level style guide is the compliance document for these systems and may be supplemented by a system-level style guide as needed.

Best Practices

The following style guide is mandated for C4ISR and combat support domain systems.

- User Interface Specification for the Defense Information Infrastructure (DII), Version 2.0, June 1996.

7.3.3.2.5 System-level Style Guides

System-level style guides provide the special tailoring of commercial, DoD, and domain-level style guides. These documents include explicit design guidance and rules for the system, while maintaining the appearance and behavior provided in the domain-level style guide. If needed, the Motif-based system-level style guide will be created in accordance with the User Interface Specification for the DII.

7.3.3.3 Symbology

The following standard is mandated for the display of common warfighting symbology:

- MIL-STD-2525A, Common Warfighting Symbology, 15 December 1996.

7.4 Operating Considerations

Often there are broad external considerations that effect the design/specification of a computing resource. This section discusses some of these considerations and how they effect the design/specification.

7.4.1 Security

Although security requirements are often considered at the end of a system design or after the system is procured and installed, much time, effort and money can be saved if security is considered first. From the computer resource point of view, security can affect the specification of the OS and some of the hardware (such as removable vs. permanent hard drives or dynamic vs. static memory). A detailed discussion of Security is provided in Chapter 3.

7.5 Communications

Almost all computing resources will require a network connection. Even if it is determined that external communications will not be required immediately, it is wise to accommodate a network interface card or modem in the computing resource design/specification. Standards play a critical role in this area and compatibility is by far the most important factor to consider. A further discussion of computer communications is provided in Chapter 5, Information Transfer and Chapter 6, Information Distribution.

7.5.1 Real Time

Often, real-time systems are thought of as being fast. The real consideration in a real-time system is meeting a process's deadline. There are hard deadlines where, if missed by the process, the data produced by the process is useless. Then there are soft deadlines where, if missed by the process, the value of the data is degraded. Sometimes these deadlines are based on human perceptions and can vary from person to person. For example, how much delay in a video teleconference is tolerable?

The scheduling of many processes on the CPU and accessing peripherals can be critical in real-time systems. When specifying OSs for systems where significant or catastrophic failures can occur when processes are not scheduled properly, consider very carefully the method by which the OS schedules processes. There should be some means of specifying the priority of processes so that a lower priority process does not prevent a higher priority process from meeting its deadlines and causing potential damage to the system, people or environment. In non real-time systems, other considerations will probably take precedence, such as cost, portability of applications, availability, or network capabilities.

Best Practices

When specifying a computer resource for a real-time system, hardware that operates at higher clock speeds (faster) is more desirable. Also use OSs that comply with ISO/IEC 9945-1: 1996:(Real-time Extensions). When considering real-time distributed systems, also refer to IEEE P1003.21-standard interfaces to real-time distributed communications.

7.5.2 High Availability

Some systems require high availability because of the high cost incurred when these systems are down (e.g. tactical systems or an organizations IT infrastructure). Systems with high availability achieve fault tolerance through redundant, distributed or protected (low vulnerability to attack) methods. High-availability systems are equipped with features that make them reliable and, therefore, fail less often than regular systems. When failure does occur, however, high-availability systems can be restored faster and more accurately than regular systems. Advanced levels of high-availability systems provide even greater capabilities:

> **Fault-resilient systems (or high-resiliency systems)** – These systems have a component of hardware detection at the computing level. Although major elements of the computer can bring down the system or portions of it, minor components can fail without interruption. Failed components—power supplies, system fans, disk drives, and some I/O controllers—can be replaced without service downtime. The fault-resilient system is expected to reduce downtime by 50 percent. The system averages less than five minutes downtimes for recovery if the design and implementation are sound.

> **Fault-tolerant systems** – These systems have redundant hardware components that continually check the other component's integrity. If one component is in error, the system

eliminates that component from the environment and notifies the operator that a failure has occurred. All failed component replacement occurs while the system is running. The component is restored to the environment with no penalty to the application. Recovery is less than one second, and all unplanned downtime is eliminated.

Continuously available systems – These systems are fault-tolerant systems that eliminate all downtime, planned and unplanned. The hardware and software can be upgraded while the system is on-line.

Some features provided by high-availability systems can be furnished through Transaction Processing (TP) monitors and some commercial database systems. Some software must be present to detect a failure and initiate the recovery process. The recovery process involves switching processing to another machine and synchronizing critical data to ensure integrity. Corruption or loss of data is determined before the processing continues. Various levels of sophistication exist, but with sophistication comes complexity and, usually, additional costs.

Some other methods that increase the availability of a system are the use of uninterruptable power supplies, RAID systems (see Section 7.2.1.5.2), systems that allow live insertion of components, distributed systems, multiple path connectivity,

Best Practices

Select system elements and design in redundancy based on the level of tolerance for unplanned downtime. Avoid single points of failure and bottlenecks in the system design. Assemble components commensurate with cost issues, functionality, and associated risk.

7.6 References

7.6.1 Standards and Specifications Resources

Computer I/O Interface Standards

American National Standards Institute (ANSI); ANSI X3.230-1994. Information Technology – Fibre Channel – Physical and Signaling Interface (FC-PH). 1994; www.nssn.org (23 May 1998)

American National Standards Institute (ANSI); ANSI X3.230-1994. Information Technology – Fibre Channel – Physical and Signaling Interface (FC-PH). 1994; www.nssn.org (23 May 1998)

American National Standards Institute (ANSI); ANSI X3.230-1994/AM 1-1996. Information Technology – Fibre Channel Physical and Signaling Interface (FC-PH) – Amendment 1. 1996; www.nssn.org (23 May 1998)

American National Standards Institute (ANSI); ANSI X3.297-1997. Information Technology – Fibre Channel – Physical and Signalling Interface-2 (FC-PH-2). 1997; www.nssn.org (23 May 1998)

American National Standards Institute (ANSI); ANSI X3.131-1994 : Information Systems - Small Computer Systems Interface-2 (SCSI-2); 1997; www.nssn.org (23 May 1998)

American National Standards Institute (ANSI); ANSI X3.253-1995 : Information Technology - SCSI-3 Parallel Interface (SPI); 1995; www.nssn.org (23 May 1998)

American National Standards Institute (ANSI); ANSI X3.232-1996 : Information Technology - SCSI-2 Common Access Method Transport and SCSI Interface Module; 1996; www.nssn.org (23 May 1998)

American National Standards Institute (ANSI); ANSI X3.269-1996 : Information Technology - Fibre Channel Protocol for SCSI; 1996; www.nssn.org (23 May 1998)

American National Standards Institute (ANSI); ANSI X3.270-1996 : Information Technology - SCSI-3 Architecture Model (SAM); 1996; www.nssn.org (23 May 1998)

American National Standards Institute (ANSI); ANSI X3.277-1996 : Information Technology - SCSI-3 Fast-20; 1996; www.nssn.org (23 May 1998)

American National Standards Institute (ANSI); ANSI X3.294-1996 : Information Technology - Serial Storage Architecture- SCSI-2 Protocol (SSA-S2P); 1996; www.nssn.org (23 May 1998)

American National Standards Institute (ANSI); ANSI/AIIM MS58-1996 : Standard Recommended Practice for Implementation of Small Computer Systems Interface (SCSI-2), (X3.131.1994) for Scanners; 1996; www.nssn.org (23 May 1998)

American National Standards Institute (ANSI); ANSI X3.276-1997 : Information Technology - SCSI-3 Controller Commands (SCC); 1997; www.nssn.org (23 May 1998)

American National Standards Institute (ANSI); ANSI X3.292-1997 : Information Technology - SCSI-3 Interlocked Protocol (SIP); 1997; www.nssn.org (23 May 1998)

American National Standards Institute (ANSI); ANSI X3.301-1997 : Information Technology - SCSI-3 Primary Commands (SPC); 1997; www.nssn.org (23 May 1998)

American National Standards Institute (ANSI); ANSI X3.304-1997 : Information Technology - SCSI-3 Multimedia Commands (MMC); 1997; www.nssn.org (23 May 1998)

American National Standards Institute (ANSI); ANSI NCITS 309-1997 : Information Technology - Serial Storage Architecture - SCSI-3 Protocol (SSA-S3P); 1997; www.nssn.org (23 May 1998)

American National Standards Institute (ANSI); ANSI NCITS 306-1998 : Information Technology - SCSI-3 Block Commands (SBC); 1998; www.nssn.org (23 May 1998)

Institute of Electrical and Electronics Engineers (IEEE) standard: 1394-1995 “IEEE Standard for a High Performance Serial Bus;” 1985; http://standards.ieee.org/db/status/ (23 May 1998)

Computer Bus Standards

Institute of Electrical and Electronics Engineers (IEEE) standard: 1014-1987 “IEEE Standard for a Versatile Backplane Bus: VMEbus;” 1987; http://standards.ieee.org/db/status/ (23 May 1998);

Institute of Electrical and Electronics Engineers (IEEE) standard: 1155-1992 “IEEE Standard VMEbus Extensions for Instrumentation: VXIbus;” http://standards.ieee.org/db/status/ (23 May 1998)

Computer OS Standards

International Organization for Standardization (ISO) standard: ISO/IEC 9945:1996, “Information Technology - Portable Operating System Interface for Computer Environments (POSIX);” 1996; http://www.iso.ch/cate/cat.html (23 May 1998)

The Open Group (TOG), X/Open CAE Specification: “System Interface Definitions, Issue 4, Version 2;” September 1994; http://www.opengroup.org/publications/catalog/c434_a.htm (23 May 1998)

The Open Group (TOG), X/Open CAE Specification: “System Interfaces and Headers, Issue 4, Version 2;” September 1994; http://www.opengroup.org/publications/catalog/c435.htm (23 May 1998)

The Open Group (TOG), X/Open CAE Specification: “Commands and Utilities, Issue 4, Version 2;” September 1994; http://www.opengroup.org/pubs/catalog/c436.htm (23 May 1998)

The Open Group (TOG): “The Single UNIX Specification” - 5 Volume Set for UNIX 95 Open Group Publication Set T910; July 1996; http://www.opengroup.org/pubs/catalog/t910s.htm (23 May 1998)

Distributed Computing Services Standards

The Open Group (TOG): “Distributed Computing Environment (DCE) Version 1.2.2;” 1997; http://www.camb.opengroup.org/tech/dce/ (24 May 1998)

Open Management Group (OMG); “The Common Object Request Broker Architecture (CORBA) and Specification Revision 2.2;”1 February 1998; http://www.omg.org/corba/corbiiop.htm (24 May 1998)

Defense Information Infrastructure Common Operating Environment (DII COE)

Defense Information Systems Agency (DISA); “Defense Information Infrastructure Common Operating Environment (DII COE);” 1 May 1998; http://spider.osfl.disa.mil/dii/ (24 May 1998)

Application Programming Interface (API)

Microsoft Inc.; Win32 APIs, Window Management and Graphics Device Interface, Volume 1 Microsoft Win32 Programmers Reference Manual; 1993, Microsoft Press, Redmond, WA

International Organization for Standardization (ISO) standard: ISO/IEC 9945-1:1996, “Information Technology - Portable Operating System Interface for Computer Environments (POSIX) - Part 1: System Application Program Interface (API)[C language];” 1996 ;http://www.iso.ch/cate/d24426.html (24 May 1998)

International Organization for Standardization (ISO) standard: ISO 9945-2: 1993, “Information Technology - Portable Operating System Interface for Computer Environments (POSIX) - Part 2: Shell and Utilities”, (as profiled by FIPS PUB 189: 1994); 1993; http://www.iso.ch/cate/d17841.html (24 May 1998)

Institute of Electrical and Electronics Engineers (IEEE) standard: IEEE 1003.2d: 1994, “POSIX - Part 2: Shell and Utilities - Amendment: Batch Environment;” 1994 http://standards.ieee.org/db/status/ (24 May 1998)

Institute of Electrical and Electronics Engineers (IEEE) standard: IEEE 1003.5 - 1992, IEEE “Standard for Information Technology - POSIX Ada Language Interfaces - Part 1: Binding for System Application Program Interface (API);” 1992; http://standards.ieee.org/db/status/ (24 May 1998)

Human Computer Interface (HCI)

Defense Information Systems Agency (DISA) Document CM-400-18-04: User Interface Specification for the Defense Information Infrastructure (DII), Version 2.0, June 1996

Department of Defense (DoD): Human Computer Interface (HCI) Style Guide”, TAFIM Version 3.0, Volume 8; 30 April 1996; http://www.library.itsi.disa.mil/org/dod_std/dod_hci_sg_v3_0.html (24 May 1998);

Smith, Mosier (MITRE); “Guidelines for Designing User Interface Software (ESD – TR-86-278)” 1996; http://www.syd.dit.csiro.au/hci/guidelines/sam/gintro.html (24 May 1998)

Open Software Foundation (OSF)/Motif™ Style Guide, Revision 1.2 (OSF 1992)

TriTeal Corporation; TriTeal Enterprise Desktop (TED) 4.0 Style Guide and Certification Checklist, TriTeal Corporation, Carlsbad, CA; 1995

Microsoft Inc.; The Windows™ Interface Guidelines For Software Design, 1996; Microsoft Press; Redmond WA

Tactical Symbology

Department of Defense; MIL-STD-2525A, “Common Warfighting Symbology;” 15 December

Real Time Requirements

Institute of Electrical and Electronics Engineers (IEEE) standard: IEEE 1003.5b - 1996, “IEEE Standard for Information Technology - POSIX Ada Language Interfaces - Part 1: Binding for System Application Programming Interface (API) - AMENDMENT 1: Real-time Extensions;” 1996; http://standards.ieee.org/db/status/ (24 May 1998)

7.6.2 Supporting Resources

Further technical information on the various aspects of the computing resource can be found at the following web sites:

Fiber Channel Association web site: http://www.fibrechannel.com (24 May 1998)

Intel Inc. iCOMP (r) web site: http://www.intel.com/procs/perf/icomp/index.htm (24 May 1998)

Microsoft Inc. web site: http://www.microsoft.com (24 May 998)

National Committee on Information Technology Standards (NCITS) T10 Technical Committee on Lower Layer Interfaces web site: http://www.symbios.com/x3t10/Welcome.html (24 May 1998)

Network Computer Consortium web site: http://www.nc.ihost.com/ (24 May 1998)

Object Management Group (OMG) web site: http://www.omg.org (24 May 1998)

PCI Special Interest Group web site: http://www.pcisig.com (24 May 1998)

Personal Computer Memory Card International Association web site: http://www.pc-card.com/ (24 May 1998)

Personal Digital Assistant (PDA) Industry Association web site: http://www.pdaia.org/ (24 May 1998)

Space and Naval Warfare Systems Command, PMW 171 web site: http://c4iweb.spawar.navy.mil (24 May 1998)

The Open Group Website: http://www.opengroup.org (24 May 1998)